# Integer Programming Modeling

IMA New Directions Short Course on Mathematical Optimization

Jeff Linderoth

Department of Industrial and Systems Engineering
Wisconsin Institutes of Discovery
University of Wisconsin-Madison

August 10, 2016

# I Say MIP It. MIP It Good.

- Today we start talking about the Mixed Integer Linear Program:

$$\text{MILP:} \quad z^* = \max\{c^\top x + h^\top y : (x, y) \in S\}$$

where

$$S := \{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : Ax + Gy \leq b\}$$

# I Say MIP It. MIP It Good.

- Today we start talking about the Mixed Integer Linear Program:

$$\text{MILP:} \quad z^* = \max\{c^\top x + h^\top y : (x, y) \in S\}$$

where

$$S := \{(x, y) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : Ax + Gy \leq b\}$$

---

- This is the The Most Important class of optimization problems

**IPs Rule!**

- Turn constraints on and off.
- Indicate whether or not constraints hold.
- Enforce logical relationships between these conditions
- Model (low-dimensional) piecewise-linear functions

# MIP for $$$

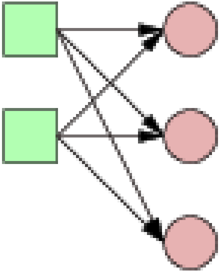
"He just immediately figured out, if you want to succeed in this business you needed a MIP."

## MIP Industries

1. Supply Chain
2. Electric Power
3. Finance
4. Work Force Management
5. Airlines
6. Railroads

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$

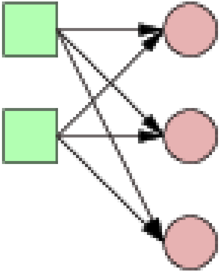


$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\begin{aligned} \sum_{i \in I} x_{ij} &= 1 \quad \forall j \in J \\ \sum_{j \in J} x_{ij} &\leq |J| y_i \quad \forall i \in I \end{aligned} \tag{1}$$

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$



$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

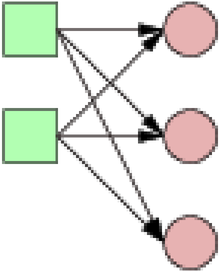
$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$$

$$\sum_{j \in J} x_{ij} \leq |J| y_i \quad \forall i \in I \qquad (1)$$

OR $x_{ij} \leq y_i \quad \forall i \in I, \ j \in J$ (2)

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$



$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$$

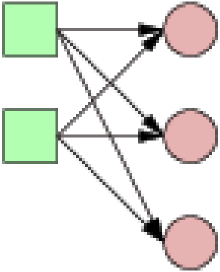
$$\sum_{j \in J} x_{ij} \leq |J| y_i \quad \forall i \in I \qquad (1)$$

OR $$x_{ij} \leq y_i \quad \forall i \in I,\ j \in J \qquad (2)$$

- Which formulation is to be preferred?

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$



$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$$

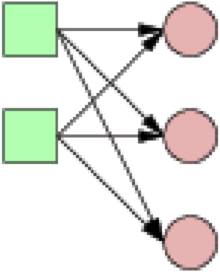
$$\sum_{j \in J} x_{ij} \leq |J| y_i \quad \forall i \in I \qquad (1)$$

$$\text{OR } x_{ij} \leq y_i \quad \forall i \in I, \ j \in J \qquad (2)$$

- Which formulation is to be preferred?
- $I = J = 40$. Costs random.

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$



$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$$

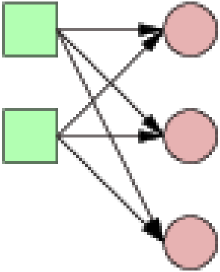
$$\sum_{j \in J} x_{ij} \leq |J| y_i \quad \forall i \in I \qquad (1)$$

OR $x_{ij} \leq y_i \quad \forall i \in I, \ j \in J \qquad (2)$

- Which formulation is to be preferred?
- $I = J = 40$. Costs random.
  - Formulation 1. 53,121 seconds, optimal solution.
  - Formulation 2. 2 seconds, optimal solution.

# Supply Chain: (Uncapacitated) Facility Location

- Facilities: $I$
- Customers: $J$



$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\begin{aligned} \sum_{i \in I} x_{ij} &= 1 \quad \forall j \in J \\ \sum_{j \in J} x_{ij} &\leq |J| y_i \quad \forall i \in I \qquad (1) \\ \text{OR } x_{ij} &\leq y_i \quad \forall i \in I,\ j \in J \qquad (2) \end{aligned}$$

- Which formulation is to be preferred?
- $I = J = 40$. Costs random.
  - Formulation 1. 53,121 seconds, optimal solution.
  - Formulation 2. 2 seconds, optimal solution.
- Why!? Stay tuned. (Next lecture)

# Supply Chain: Lot Sizing

- $x_t$: Production quantity in period $t$
- $s_t$: Inventory at end of period $t$
- $y_t$: Production indicator in period $t$

---

# Supply Chain: Lot Sizing

- $x_t$: Production quantity in period $t$
- $s_t$: Inventory at end of period $t$
- $y_t$: Production indicator in period $t$

---

$$
\begin{aligned}
\min \ & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t. } & s_{t-1} + x_t - s_t = d_t, \quad t = 1, \dots, n \\
& x_t \leq D_{tn} y_t, \qquad t = 1, \dots, n \\
& s_0 = s_n = 0, s_t \geq 0, x_t \geq 0, y_t \in \{0, 1\}, \ t = 1, \dots, n
\end{aligned}
$$

where $D_{j\ell} = \sum_{t=j}^{\ell} d_t$ for $j \leq \ell$.

---

# Supply Chain: Lot Sizing

- $x_t$: Production quantity in period $t$
- $s_t$: Inventory at end of period $t$
- $y_t$: Production indicator in period $t$

---

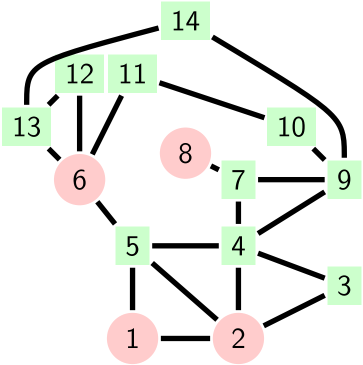
$$
\begin{aligned}
\min \ & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t. } & s_{t-1} + x_t - s_t = d_t, \quad t = 1, \ldots, n \\
& x_t \leq D_{tn} y_t, \qquad t = 1, \ldots, n \\
& s_0 = s_n = 0, s_t \geq 0, x_t \geq 0, y_t \in \{0, 1\}, \ t = 1, \ldots, n
\end{aligned}
$$

where $D_{j\ell} = \sum_{t=j}^{\ell} d_t$ for $j \leq \ell$.

---

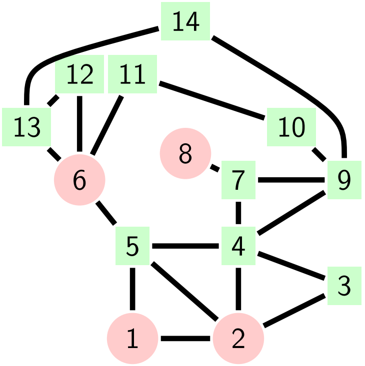
- This is our "working IP" for the exercises

# Power Systems




- Power Network: $(N, A)$ with...
- $G \subset N$: generation nodes
- $D \subset N$: demand nodes
- Load forecasts (MW) $b_i$ for $i \in D$
- Generation cost (\$/MW) $c_i$ and capability $\overline{p}_i$ (MW) for $i \in G$
- Peak load rating (MW) $u_{ij}$ for $(i, j) \in A$

# Power Systems




- Power Network: $(N, A)$ with...
- $G \subset N$: generation nodes
- $D \subset N$: demand nodes
- Load forecasts (MW) $b_i$ for $i \in D$
- Generation cost ($/MW) $c_i$ and capability $\bar{p}_i$ (MW) for $i \in G$
- Peak load rating (MW) $u_{ij}$ for $(i, j) \in A$

## Economic Dispatch Problem

- Determine power generation levels for $i \in G$ and power transmission levels for $(i, j) \in A$ to meet demands $b_i, i \in D$, at minimum cost

# Power Flow

- Electric power grids follow the laws of physics, characterized by nonlinear, nonconvex equations
- DC Power Flow Approximation: The (real) power $x_{ij}$transmit over line $(i, j) \in A$ is proportional to angle differences $(\theta_i, \theta_j)$ at the endpoint nodes:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j)$$

- $p_i$: (Real) power inject at generator $i \in G$

# Linear Program for (DC) Economic Dispatch

$$\min_{x,p,\theta} \sum_{i \in G} c_i p_i$$

s.t.

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} p_i & \forall i \in G \\ d_i & \forall i \in D \\ 0 & \forall i \in N \setminus G \setminus D \end{cases}$$

$$-u_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E$$

$$\underline{p}_i \leq p_i \leq \overline{p}_i \quad \forall i \in G$$

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \quad \forall (i,j) \in E$$

$$x_{ij} \in \mathbb{R} \quad \forall (i,j) \in E$$

$$p_i \in \mathbb{R}_+ \quad \forall i \in G$$

$$\theta_i \in \mathbb{R} \quad \forall i \in N$$

# MCNF++

- This economic dispatch problem is just a min cost network flow problem with some additional "potential" constraints

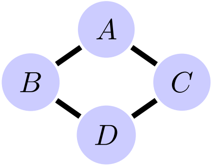
$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j)$$

# MCNF++

- This economic dispatch problem is just a min cost network flow problem with some additional "potential" constraints

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j)$$

- Implication: The potential drop $(\theta_A - \theta_D)$ must be the same along the paths: $A \to B \to D$ and $A \to C \to D$

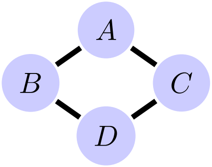


# MCNF++

- This economic dispatch problem is just a min cost network flow problem with some additional "potential" constraints

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j)$$

- Implication: The potential drop $(\theta_A - \theta_D)$ must be the same along the paths: $A \to B \to D$ and $A \to C \to D$




## "Braess Paradox"

- If line $(C, D)$ didn't exist, I wouldn't have to enforce this potential balance constraint.
- Thus, removing lines of the transmission network may actually increase the efficiency of delivery.

# Transmission Switching

## Tradeoff

- Good: Having Lines Allows You to Send Flow:

$$-U_{ij} \leq x_{ij} \leq U_{ij} \ \forall (i,j) \in E$$

- Bad: Having Lines Induces Constraints in the Network:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \ \forall (i,j) \in E$$

# Transmission Switching

## Tradeoff

- Good: Having Lines Allows You to Send Flow:

$$-U_{ij} \leq x_{ij} \leq U_{ij} \ \forall (i,j) \in E$$

- Bad: Having Lines Induces Constraints in the Network:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \ \forall (i,j) \in E$$

---

- Fisher et al. [2008] show that efficiency improved by switching off transmission lines

| **Lines Off** | % Improvement |
|---|---|
| 1 | 6.3% |
| 2 | 12.4% |
| 3 | 19.9% |
| 4 | 20.5% |
| [illegible] | [illegible] |

# Switching Off Lines

- Regular Flow Constraints:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \quad \forall (i,j) \in E$$
$$-U_{ij} \leq x_{ij} \leq U_{ij} \quad \forall (i,j) \in E$$

# Switching Off Lines

- Regular Flow Constraints:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \quad \forall (i,j) \in E$$
$$-U_{ij} \leq x_{ij} \leq U_{ij} \quad \forall (i,j) \in E$$

- Let $z_{ij} \in \{0,1\} \ \forall (i,j) \in A$
- Switched Flow Constraints:

$$x_{ij} = \alpha_{ij} z_{ij} (\theta_i - \theta_j) \quad \forall (i,j) \in E$$

# Switching Off Lines

- Regular Flow Constraints:

$$x_{ij} = \alpha_{ij}(\theta_i - \theta_j) \quad \forall (i,j) \in E$$
$$-U_{ij} \le x_{ij} \le U_{ij} \quad \forall (i,j) \in E$$

- Let $z_{ij} \in \{0,1\} \ \forall (i,j) \in A$
- Switched Flow Constraints:

$$x_{ij} = \alpha_{ij} z_{ij} (\theta_i - \theta_j) \quad \forall (i,j) \in E$$

## That is NOT a MILP!

- If (and only if) $\theta_i$ have bounds then one can write an MILP formulation
- $z_{ij} = 1 \Leftrightarrow$ line $(i,j) \in A$ is used

# MILP Formulation

$$\min_{x,p,\theta,z} \quad \sum_{i \in G} c_i p_i$$

s.t.

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ij} = \begin{cases} p_i & \forall i \in G \\ d_i & \forall i \in D \\ 0 & \forall i \in N \setminus G \setminus D \end{cases}$$

$$-U_{ij} z_{ij} \le x_{ij} \le U_{ij} z_{ij} \quad \forall (i,j) \in E$$

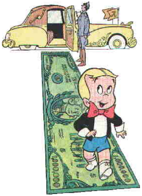
$$\alpha_{ij}(\theta_i - \theta_j) - x_{ij} + M(1 - z_{ij}) \ge 0 \quad \forall (i,j) \in E$$

$$\alpha_{ij}(\theta_i - \theta_j) - x_{ij} - M(1 - z_{ij}) \le 0 \quad \forall (i,j) \in E$$

$$-L_i \le \theta_i \le L_i \quad \forall i \in N$$

$$\underline{p}_i \le p_i \le \overline{p}_i \quad \forall i \in G$$

$$z_{ij} \in \{0,1\} \quad \forall (i,j) \in E$$

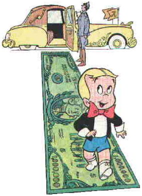


# Portfolio Management

- $N$: Universe of asset to purchase
- $x_i$: Amount of asset $i$ to hold
- $B$: Budget

$$\min_{x \in \mathbb{R}_+^{|N|}} \left\{ u(x) \mid \sum_{i \in N} x_i = B \right\}$$

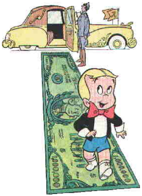
# Portfolio Management



- $N$: Universe of asset to purchase
- $x_i$: Amount of asset $i$ to hold
- $B$: Budget

$$\min_{x \in \mathbb{R}_+^{|N|}} \left\{ u(x) \mid \sum_{i \in N} x_i = B \right\}$$

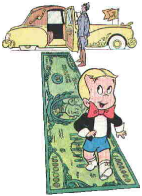
- Markowitz: $u(x) \stackrel{\text{def}}{=} -\alpha^T x + \lambda x^T Q x$
  - $\alpha$: Expected returns
  - $Q$: Variance-covariance matrix of expected returns
  - $\lambda$: Risk aversion parameter



# More Realistic Models

- $b \in \mathbb{R}^{|N|}$ of "benchmark" holdings
- Benchmark Tracking: $u(x) \stackrel{\text{def}}{=} (x-b)^T Q (x-b)$
  - Constraint on $\mathbb{E}$[Return]: $\alpha^T x \geq r$

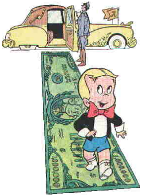
# More Realistic Models



- $b \in \mathbb{R}^{|N|}$ of "benchmark" holdings
- Benchmark Tracking: $u(x) \stackrel{\text{def}}{=} (x-b)^T Q(x-b)$
  - Constraint on $\mathbb{E}$[Return]: $\alpha^T x \geq r$
- Limit Names: $|i \in N \ : \ x_i > 0| \leq K$
  - Use binary indicator variables to model the implication $x_i > 0 \Rightarrow y_i = 1$
  - Implication modeled with variable upper bounds:

$$x_i \leq B y_i \qquad \forall i \in N$$

  - $\sum_{i \in N} y_i \leq K$

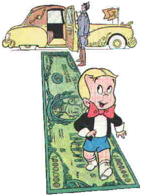


# Even More Models

**Min Holdings: $(x_i = 0) \vee (x_i \geq m)$**

- Model implication: $x_i > 0 \Rightarrow x_i \geq m$
- $x_i > 0 \Rightarrow y_i = 1 \Rightarrow x_i \geq m$
- $x_i \leq By_i, x_i \geq my_i \ \forall i \in N$

# Even More Models



## Min Holdings: $(x_i = 0) \vee (x_i \geq m)$

- Model implication: $x_i > 0 \Rightarrow x_i \geq m$
- $x_i > 0 \Rightarrow y_i = 1 \Rightarrow x_i \geq m$
- $x_i \leq By_i, x_i \geq my_i \ \forall i \in N$

## Round Lots: $x_i \in \{kL_i, k = 1, 2, \ldots\}$

- $x_i - z_iL_i = 0, z_i \in \mathbb{Z}_+ \ \forall i \in N$

# Modeling Complicated Logical Relationships

- It is often quite difficult[1] to quickly see how one can write complictaed logical conditions as algebraic conditions (that can then be implemented in an algebraic modeling language like AMPL)

---

[1] At least for my puny brain

# Modeling Complicated Logical Relationships

- It is often quite difficult[1] to quickly see how one can write complictaed logical conditions as algebraic conditions (that can then be implemented in an algebraic modeling language like AMPL)

---

- What follows is the derivation of a kind of “calculus” for automatically converting logical conditions into equivalent algebraic ones

---

[1] At least for my puny brain

# Modeling Complicated Logical Relationships

- It is often quite difficult[1] to quickly see how one can write complictaed logical conditions as algebraic conditions (that can then be implemented in an algebraic modeling language like AMPL)

---

- What follows is the derivation of a kind of "calculus" for automatically converting logical conditions into equivalent algebraic ones

---

## Examples for UFL

1. If you open $k$ or more facilities, then you must pay a penalty cost of $\lambda$
2. If you open facility one or two, then you may not open both facility 3 and 4
3. If facility 1 and 2 are both open, then customer 3 must get all of his demand from at most 1 facility

[1] At least for my puny brain

# A "Calculus" for Logical Modeling

## Variable $= 1 \Rightarrow$ constraint must be satisfied

- Suppose we wish to have a constraint hold if an associated indicator variable $\delta$ is flipped to 1. That is...

$$\boxed{\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b}$$

# A "Calculus" for Logical Modeling

**Variable $= 1 \Rightarrow$ constraint must be satisfied**

- Suppose we wish to have a constraint hold if an associated indicator variable $\delta$ is flipped to 1. That is...

$$\boxed{\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b}$$

- This can be represented by the constraint
  - $\sum_{j \in N} a_j x_j + M\delta \leq M + b$
  - $M$ is an upper bound for the expression $\sum_{j \in N} a_j x_j - b$.

# The Logic

$$\boxed{\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b \Leftrightarrow \sum_{j \in N} a_j x_j + M\delta \leq M + b}$$

- Equivalent to $\sum_{j \in N} a_j x_j - b \leq M(1 - \delta)$
- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j - b \leq M$
    - (true by definition of $M$)
- $\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j - b \leq 0$

# Modeling Trick #2: Converse of First

$$\boxed{\sum_{j\in N} a_j x_j \leq b \Rightarrow \delta = 1}$$

- $\delta = 0 \Rightarrow \sum_{j\in N} a_j x_j \not\leq b$
- $\delta = 0 \Rightarrow \sum_{j\in N} a_j x_j > b$
- $\delta = 0 \Rightarrow \sum_{j\in N} a_j x_j \geq b + \epsilon$

# Modeling Trick #2: Converse of First

$$\boxed{\sum_{j \in N} a_j x_j \leq b \Rightarrow \delta = 1}$$

- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j \not\leq b$
- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j > b$
- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j \geq b + \epsilon$
    - If $a_j, x_j$ are integer, we can choose $\epsilon = 1$

# Modeling Trick #2: Converse of First

$$\boxed{\sum_{j \in N} a_j x_j \leq b \Rightarrow \delta = 1}$$

- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j \not\leq b$
- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j > b$
- $\delta = 0 \Rightarrow \sum_{j \in N} a_j x_j \geq b + \epsilon$
    - If $a_j, x_j$ are integer, we can choose $\epsilon = 1$
- Model as $\sum_{j \in N} a_j x_j - (m - \epsilon)\delta \geq b + \epsilon$
    - $m$ is a lower bound for the expression $\sum_{j \in N} a_j x_j - b$
- $\delta = 0 : \sum_{j \in N} a_j x_j \geq b + \epsilon$
- $\delta = 1 : m \leq \sum_{j \in N} a_j x_j - b$ (nothing)

# Some Last Modeling Tricks

$$\boxed{\delta = 1 \Rightarrow \sum_{j\in N} a_j x_j \geq b}$$

- Model as $\sum_{j\in N} a_j x_j + m\delta \geq m + b$

$$\boxed{\sum_{j\in N} a_j x_j \geq b \Rightarrow \delta = 1}$$

- Model as $\sum_{j\in N} a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$

# The Slide of Tricks

- $\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b$
  - $\sum_{j \in N} a_j x_j + M\delta \leq M + b$
- $\sum_{j \in N} a_j x_j \leq b \Rightarrow \delta = 1$
  - $\sum_{j \in N} a_j x_j - (m - \epsilon)\delta \geq b + \epsilon$
- $\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \geq b$
  - $\sum_{j \in N} a_j x_j + m\delta \geq m + b$
- $\sum_{j \in N} a_j x_j \geq b \Rightarrow \delta = 1$
  - $\sum_{j \in N} a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$

- $\sum_{j \in N} a_j x_j$: Constraint LHS
- $b$: Constraint RHS
- $M$: Upper bound on $\sum_{j \in N} a_j x_j - b$
- $m$: Lower bound on $\sum_{j \in N} a_j x_j - b$
- $\epsilon$: Constraint violation amount ($\epsilon = 0.01 or 1$)

# Simple Tricks

- These two tricks are very common and can be derived from the “Slide of Tricks”
- Variable Upper Bound: $x > 0 \Rightarrow \delta = 1$
  - $x \leq M\delta$
- Variable Lower Bound: $\delta = 1 \Rightarrow x \geq m$
  - $x \geq m\delta$

# Recall UFL

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

subject to

$$\begin{aligned} x_{ij} &\leq y_i && \forall i \in I, \forall j \in J \\ \sum_{i \in I} x_{ij} &= 1 && \forall j \in J \\ x_{ij} &\geq 0 && \forall i \in I, \forall j \in J \\ y_i &\in \{0, 1\} && \forall i \in I \end{aligned}$$

# UFL 1

- If you open $k$ or more facilities, then you must pay a penalty cost of $\lambda$

# UFL 1

- If you open $k$ or more facilities, then you must pay a penalty cost of $\lambda$

---

- Need (earlier) "fixed cost" logic for $y_i$ : $(x_{ij} \leq y_i)$
- Model $\sum_{i \in M} y_i \geq k \Rightarrow \delta_1 = 1$
- Add $\lambda\delta_1$ to objective function

---

# UFL 1

- If you open $k$ or more facilities, then you must pay a penalty cost of $\lambda$

---

- Need (earlier) "fixed cost" logic for $y_i$ : $(x_{ij} \leq y_i)$
- Model $\sum_{i \in M} y_i \geq k \Rightarrow \delta_1 = 1$
- Add $\lambda \delta_1$ to objective function

---

- Appropriate trick is

$$\sum_{j \in N} a_j x_j \geq b \Rightarrow \delta = 1 \Leftrightarrow \sum_{j \in N} a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$$

- $M = |I| - k$. $\epsilon = 1$:

$$\sum_{i \in I} y_i - (|I| - k + 1)\delta_1 \leq k - 1.$$

# UFL 2

- If you open facility one or two, then you may not open both facility 3 and 4

# UFL 2

- If you open facility one or two, then you may not open both facility 3 and 4

---

- Then need to model $y_1 + y_2 \geq 1 \Rightarrow y_3 + y_4 \leq 1$
  - $y_1 + y_2 \geq 1 \Rightarrow \delta_2 = 1$
  - $\delta_2 = 1 \Rightarrow y_3 + y_4 \leq 1$

# UFL 2, cont.

- First trick is:

$$\sum_{j \in N} a_j x_j \geq b \Rightarrow \delta = 1 \Leftrightarrow \sum_{j \in N} a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$$

$M = 1$, $\epsilon = 1$:

$$y_1 + y_2 - 2\delta_2 \leq 0.$$

(Note: could also model (better) as $\delta_2 \geq y_1, \delta_2 \geq y_2$)

- Second trick is:

$$\delta = 1 \Rightarrow \sum_{j \in N} a_j x_j \leq b \Leftrightarrow \sum_{j \in N} a_j x_j + M\delta \leq M + b$$

$M = 1, \epsilon = 1$:

$$y_3 + y_4 + \delta_2 \leq 2.$$

# UFL 3

- If facility 1 and 2 are both open, then customer 3 must get all of his demand from at most 1 facility

# UFL 3

- If facility 1 and 2 are both open, then customer 3 must get all of his demand from at most 1 facility

---

- Need indicator variables $x_{ij} > 0 \Rightarrow z_{ij} = 1$ $(x_{ij} \leq z_{ij} \forall i \in I, j \in J)$
- Then model $y_1 + y_2 \geq 2 \Rightarrow \sum_{i \in I} z_{i3} \leq 1$
  - $y_1 + y_2 \geq 2 \Rightarrow \delta_3 = 1$
  - $\delta_3 = 1 \Rightarrow \sum_{i \in I} z_{i3} \leq 1$

# UFL 3

- First trick (for $y_1 + y_2 \geq 2 \Rightarrow \delta_3 = 1$) is $\sum_{j \in N} a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$
- $M = 0$, $\epsilon = 1$:

$$y_1 + y_2 - \delta_3 \leq 1$$

- Second trick (for $\delta_3 = 1 \Rightarrow \sum_{i \in I} z_{i3} \leq 1$) is $\sum_{j \in N} a_j x_j + M\delta \leq M + b$
- $M = |I| - 1$

$$\sum_{i \in I} z_{i3} + (|I| - 1)\delta_3 \leq |M|$$

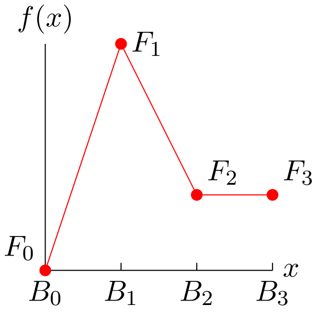
# Piecewise Linear Functions

- A very common structure used in modeling is a piecewise-linear function of a scalar variable $x \in [B_0, B_n]$

$$f(x) = m_i x + c_i, \quad x \in [B_{i-1}, B_i] \quad \forall i = 1, \ldots, n$$

## Sample applications using PLFs

- Gas network optimization [Martin et al., 2006]
- Transmissions expansion planning [Alguacil et al., 2003]
- Oil field development [Gupta and Grossmann, 2012]
- Hydro Scheduling [Borghetti et al., 2008]
- Thermal unit commitment [Carrion and Arroyo, 2006]
- Sales resource allocation [Lodish, 1971]



# Modeling Piecewise-Linear Functions

- There are many ways to model piecewise linear functions using integer variables. I really recommend Vielma et al. [2010], Vielma [2015] as references

## Take Your Pick

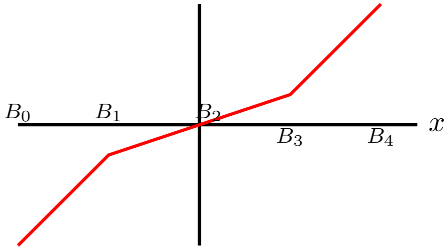
- Multiple Choice Model [Jeroslow and Lowe, 1984]
- SOS2 Model [Beale and Tomlin, 1970, Beale and Forrest, 1976]
- Incremental Model [Markowitz and Manne, 1957]
- Convex Combination Model [Dantzig, 1960, Padberg, 2000]
- Disaggregated Convex Combination Model [Meyer, 1976]
- Logarithmic Model [Vielma et al., 2010]

# Multiple Choice Model

- Want to model $(\forall i)$

$$f(x) = m_i x + c_i, \;\; x \in [B_{i-1}, B_i]$$

$$f(x) \approx \mathrm{sgn}(x)|x|^\gamma$$

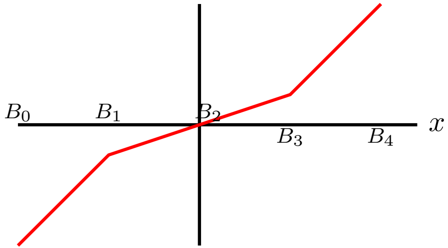


# Multiple Choice Model

- Want to model $(\forall i)$

$$f(x) = m_i x + c_i, \;\; x \in [B_{i-1}, B_i]$$

$$f(x) \approx \operatorname{sgn}(x)|x|^\gamma$$




## Introduce New Variables

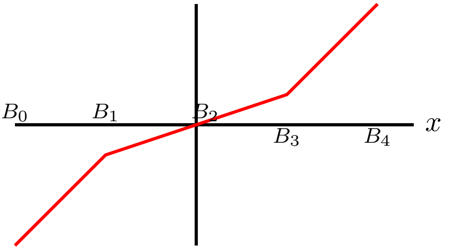
- $b_i := 1$ if $x \in [B_{i-1}, B_i]$
- $w_i := x$ if $x \in [B_{i-1}, B_i]$
- $t \approx f(x)$

# Multiple Choice Model

- Want to model $(\forall i)$

$$f(x) = m_i x + c_i, \;\; x \in [B_{i-1}, B_i]$$

$$f(x) \approx \text{sgn}(x)|x|^\gamma$$




## Introduce New Variables

- $b_i := 1$ if $x \in [B_{i-1}, B_i]$
- $w_i := x$ if $x \in [B_{i-1}, B_i]$
- $t \approx f(x)$

$$x = \sum_{i=1}^{n} w_i,$$

$$t = \sum_{i=1}^{n} (m_i w_i + c_i b_i)$$

$$B_{i-1} b_i \leq w_i \leq B_i b_i \qquad \forall i \in [n]$$

$$1 = \sum_{i=1}^{n} b_i$$

# But Wait, There's More

- In many applications mentioned there is an additional binary *indicator variable*

[2] The way done in all the papers

# But Wait, There's More

- In many applications mentioned there is an additional binary *indicator variable*
- If $y = 0$, then $x = 0$, and we would like the relationship $t = f(x)$ "turned off"
- Specifically (assuming WLOG $f(0) = 0$)

$$y = 0 \Rightarrow x = 0, \ t = f(x) = 0$$

[2] The way done in all the papers

# But Wait, There's More

- In many applications mentioned there is an additional binary *indicator variable*
- If $y = 0$, then $x = 0$, and we would like the relationship $t = f(x)$ "turned off"
- Specifically (assuming WLOG $f(0) = 0$)

$$y = 0 \Rightarrow x = 0, \ t = f(x) = 0$$

- The standard way[2] is to introduce a "big-M" constraint:

$$x \leq B_n y$$

[2] The way done in all the papers

# A Simple Trick

- Instead of modeling the piecewise-linear indicator in the standard way:

$$B_0 y \leq x \leq B_n y, \qquad 1 = \sum_{i=1}^{n} b_i$$

# A Simple Trick

- Instead of modeling the piecewise-linear indicator in the standard way:

$$B_0 y \leq x \leq B_n y, \qquad 1 = \sum_{i=1}^{n} b_i$$

- Model it as

$$y = \sum_{i=1}^{n} b_i$$

# A Simple Trick

- Instead of modeling the piecewise-linear indicator in the standard way:

$$B_0 y \le x \le B_n y, \qquad 1 = \sum_{i=1}^{n} b_i$$

- Model it as

$$y = \sum_{i=1}^{n} b_i$$

---

- Resulting formulation is provably stronger (locally-ideal)
- All piecewise-linear functions turned on/off by an indicator have a similar modeling trick Sridhar et al. [2013]
- It is the exact extension of the "perspective reformulation" (which you will learn in Friday lecture) to this case

# Multiple Choice Model

## Not "Locally ideal"

$$x = \sum_{i=1}^{n} w_i,$$

$$t = \sum_{i=1}^{n} (m_i w_i + c_i b_i),$$

$$B_{i-1} b_i \leq w_i \leq B_i b_i, \qquad \forall i \in [n]$$

$$x \leq B_n y,$$

$$1 = \sum_{i=1}^{n} b_i$$

- A formulation is locally ideal if every extreme point of the LP relaxation satisfies the discrete requirements

# Multiple Choice Model

**Not "Locally ideal"**

$$x = \sum_{i=1}^{n} w_i,$$

$$t = \sum_{i=1}^{n} (m_i w_i + c_i b_i),$$

$$B_{i-1} b_i \le w_i \le B_i b_i, \qquad \forall i \in [n]$$

$$x \le B_n y,$$

$$1 = \sum_{i=1}^{n} b_i$$

**"Locally Ideal"**

$$x = \sum_{i=1}^{n} w_i,$$

$$t = \sum_{i=1}^{n} (m_i w_i + c_i b_i),$$

$$B_{i-1} b_i \le w_i \le B_i b_i, \qquad \forall i \in [n]$$

$$y = \sum_{i=1}^{n} b_i$$

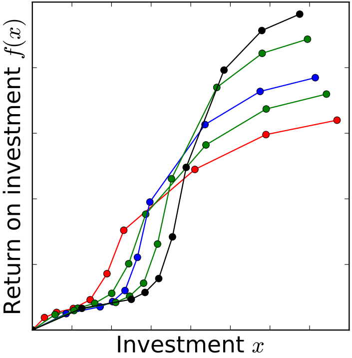
- A formulation is locally ideal if every extreme point of the LP relaxation satisfies the discrete requirements

# Ad Budget Allocation ([Zoltners and Sinha, 1980])

- Allocate advertising budget $D$ among a set $\mathcal{K}$ of advertising strategies for a set of $\mathcal{J}$ products.

---

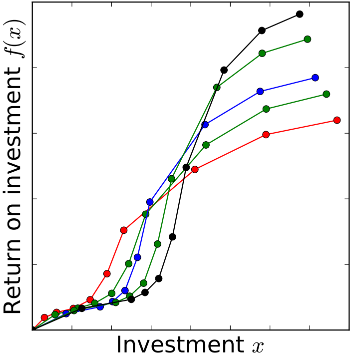
- Fixed cost $G_j$ for entering the market with product $j \in \mathcal{J}$.
- Variable cost $c_{jk}$ for each unit of the resource allocated to strategy $k \in \mathcal{K}$ of product $j \in \mathcal{J}$.
- Return on investment $f_{jk}(x_{jk})$ having $S$-curve type shape



# Ad Budget Allocation ([Zoltners and Sinha, 1980])

- Allocate advertising budget $D$ among a set $\mathcal{K}$ of advertising strategies for a set of $\mathcal{J}$ products.

- Fixed cost $G_j$ for entering the market with product $j \in \mathcal{J}$.
- Variable cost $c_{jk}$ for each unit of the resource allocated to strategy $k \in \mathcal{K}$ of product $j \in \mathcal{J}$.
- Return on investment $f_{jk}(x_{jk})$ having $S$-curve type shape




## Variables

- $x_{jk}$: advertising resource allocated to strategy $k \in \mathcal{K}$ for product $j \in \mathcal{J}$.
- $t_{jk} = f_{jk}(x_{jk})\ \forall j \in \mathcal{K}, \forall k \in \mathcal{K},\ y_j \in \{0,1\} \forall j \in J$

# MIP Formulation

$$\max \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} t_{jk}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} c_{jk} x_{jk} + \sum_{i \in \mathcal{J}} G_j y_j \leq D$$

$$(x_{jk}, t_{jk}, y_j) \in X_{jk} \quad \forall j \in \mathcal{J}, k \in \mathcal{K}.$$

- You must model a PW-Linear function for each $j \in \mathcal{J}, k \in \mathcal{K}$

$$X_{jk} := \{(x, t, y) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \{0, 1\} \mid y = 1 \Rightarrow t = f_{jk}(x),\ y = 0 \Rightarrow x = t = 0\}$$

# MIP Formulation

$$\max \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} t_{jk}$$

$$\sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} c_{jk} x_{jk} + \sum_{i \in \mathcal{J}} G_j y_j \leq D$$

$$(x_{jk}, t_{jk}, y_j) \in X_{jk} \quad \forall j \in \mathcal{J}, k \in \mathcal{K}.$$

- You must model a PW-Linear function for each $j \in \mathcal{J}, k \in \mathcal{K}$

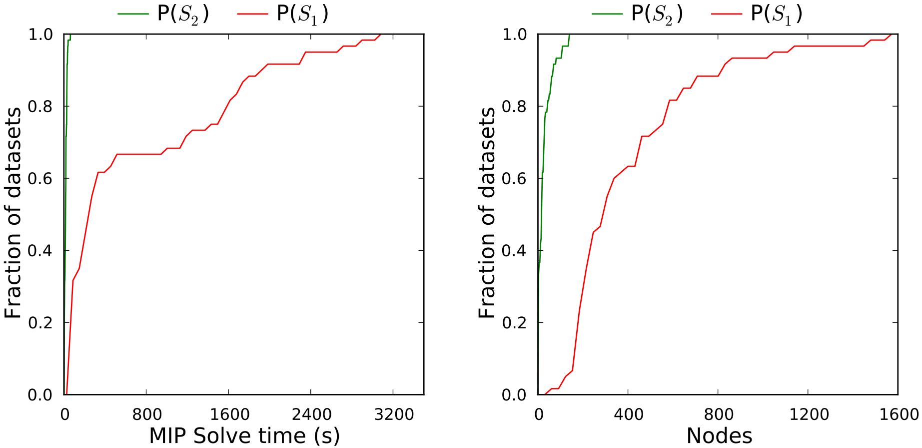
$$X_{jk} := \{(x, t, y) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \{0, 1\} \mid y = 1 \Rightarrow t = f_{jk}(x),\ y = 0 \Rightarrow x = t = 0\}$$

- How well does it work, you ask?

# Ta Da!

| | Standard Formulation | New Formulation |
|---|---|---|
| Avg. Time | 703 sec. | 17 sec. |
| Max Time | $>$3600 sec. | 509 sec. |
| Avg. Nodes | 402.9 | 26.3 |
| Max Nodes | $>$1572 | 140 |

# CDFs of nodes/times



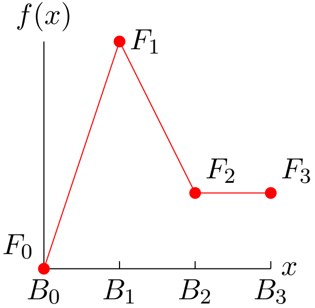
# SOS2 formulation for $t = f(x)$

$$x = \sum_{i=0}^{n} B_i \lambda_i$$

$$t = \sum_{i=0}^{n} F_i \lambda_i$$

$$1 = \sum_{i=0}^{n} \lambda_i$$

$\lambda$ is SOS2




SOS2: Special Ordered Set of Type 2

- At most two nonzeros, must be adjacent
- (Some) solvers will enforce

# Strengthening is Easy-Peasy

## Standard Formulation

$$x = \sum_{i=0}^{n} B_i \ \lambda_i,$$

$$t = \sum_{i=0}^{n} F_i \ \lambda_i$$

$$1 = \sum_{i=0}^{n} \lambda_i,$$

$\lambda$ is SOS2

$x \leq B_n y$

# Strengthening is Easy-Peasy

## Standard Formulation

$$x = \sum_{i=0}^{n} B_i \lambda_i,$$

$$t = \sum_{i=0}^{n} F_i \lambda_i$$

$$1 = \sum_{i=0}^{n} \lambda_i,$$

$\lambda$ is SOS2

$x \leq B_n y$

## Stronger and More Compact

$$x = \sum_{i=0}^{n} B_i \lambda_i$$

$$t = \sum_{i=0}^{n} F_i \lambda_i$$

$$y = \sum_{i=0}^{n} \lambda_i,$$

$\lambda$ is SOS2

# Economies of Scale: Example

- We need to buy $b$ items from a set of suppliers $I$.
  - $B$: Set of Cost Breakpoints
  - $c_{ib}$: Per Units cost of item from supplier $i \in I$ in cost region $b \in B$
  - $v_{ib}$: Maximum number of item from supplier $i \in I$ to purchase in region $b \in B$
  - $\alpha_i$: Maximum percentage to purchase from any supplier

# Economies of Scale: Example

- We need to buy $b$ items from a set of suppliers $I$.
  - $B$: Set of Cost Breakpoints
  - $c_{ib}$: Per Units cost of item from supplier $i \in I$ in cost region $b \in B$
  - $v_{ib}$: Maximum number of item from supplier $i \in I$ to purchase in region $b \in B$
  - $\alpha_i$: Maximum percentage to purchase from any supplier

---

- The adjacency conditions of SOS2 are enforced by the solution algorithm
- Commercial solvers allow you to specify SOS2.
- In AMPL, the translation is done automatically for you if you use the special piecewise notation. (See Chapter 17 of AMPL Book)

# Example:

```
table  COST()
COST 1   2   3
1    9.2 9   7
2    9   8.5 8.3
3    11  8.5 7.5
;

table  BR()
BR 0 1   2   3
1  0 100 200 1000
2  0 50  250 2000
3  0 100 300 4000
;
```

# Model

## Variables

- $x_i$: Amount to purchase from supplier $i \in I$
- $\lambda_{ib}$: Convex combination multipliers for each $x_i \in I$

# Model

## Variables

- $x_i$: Amount to purchase from supplier $i \in I$
- $\lambda_{ib}$: Convex combination multipliers for each $x_i \in I$

$$\min \sum_{i \in I} \sum_{b \in B} \gamma_{ib} \lambda_{ib}$$

$$\sum_{b \in B} v_{ib} \lambda_{ib} = x_i \quad \forall i \in I$$

$$\sum_{b \in B} \lambda_{ib} = 1 \quad \forall i \in I$$

$$x_i \leq \alpha_i b \quad \forall i \in I$$

$$\sum_{i \in I} x_i \geq b$$

$$\lambda_{ib} \text{ SOS2} \quad \forall i \in I \qquad x_i \geq 0 \quad \forall i \in I$$

AMPL Code

```
set I;

param cost1{i in I} >= 0;
param cost2{i in I} >= 0;
param cost3{i in I} >= 0;

param limit1{i in I};
param limit2{i in I} >= limit1[i];
param b;
param alpha{I} >= 0;

var x{i in I} >= 0, <= alpha[i] * b;

minimize Cost: sum{i in I} << limit1[i], limit2[i];
         cost1[i], cost2[i], cost3[i]>> x[i];

subject to BuyEnough:
  sum{i in I} x[i] >= b;
```

# Conclusions

- Integer programming is everywhere!
- Learned a "calculus" for turning logical restrictions into algebraic decriptions
- Introduction to modeling piecewise linear functions.
  - Including how to strengthen PW-linear with "on off" constraints

Natalia Alguacil, Alexis L. Motto, and Antonio J. Conejo. Transmission expansion planning: a mixed-integer LP approach. *IEEE Trans. Power Systems*, 18:1070–1077, 2003.

E. M. L. Beale and J. J. H. Forrest. Global optimization using special ordered sets. *Mathematical Programming*, 10:52–69, 1976.

E. W. L. Beale and J. A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence, editor, *Proceedings of the 5th International Conference on Operations Research*, pages 447–454, 1970.

Alberto Borghetti, Senior Member, Claudia D Ambrosio, Andrea Lodi, and Silvano Martello. An MILP Approach for Short-Term Hydro Scheduling and Unit Commitment With Head-Dependent Reservoir. *IEEE Trans. Power Systems*, 23(3):1115–1124, 2008.

M Carrion and J M Arroyo. A Computationally Efficient Mixed-Integer Linear Formulation for the Thermal Unit Commitment Problem. *IEEE Trans. Power Systems*, 21(3):1371–1378, 2006. ISSN 08858950. doi: 10.1109/TPWRS.2006.876672. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1664974`.

George Bernard Dantzig. On the significance of solving linear programming problems with some integer variables. *Econometrica*, 28:30–44, 1960.

E. B. Fisher, R. P. O'Neill, and M.C. Ferris. Optimal transmission switching. *IEEE Transactions on Power Systems*, pages 1346–1355, 2008.

Vijay Gupta and Ignacio E. Grossmann. An Efficient Multiperiod MINLP Model for Optimal Planning of Offshore Oil and Gas Field Infrastructure. *Ind. & Eng. Chem. Res.*, 51(19): 6823–6840, May 2012. ISSN 0888-5885. doi: 10.1021/ie202959w. URL `http://pubs.acs.org/doi/abs/10.1021/ie202959w`.

R. G. Jeroslow and J. K. Lowe. Modeling with integer variables. *Mathematical Programming Studies*, 22:167–184, 1984.

LM Lodish. CALLPLAN: An interactive salesman's call planning system. *Manag. Sci.*, 18(4), 1971. URL http://mansci.journal.informs.org/content/18/4-Part-II/P-25.short.

Harry M. Markowitz and Alan S. Manne. On the solution of discrete programming problems. *Econometrica*, 25(1):84–110, 1957.

Alexander Martin, Markus Möller, and Susanne Moritz. Mixed Integer Models for the Stationary Case of Gas Network Optimization. *Math. Program.*, 105:563–582, 2006.

R. R. Meyer. Integer and mixed-integer programming models: General properties. *Journal of Optimization Theory and Applications*, 16:191–206, 1976.

M. Padberg. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, 27(1):1–5, 2000.

S. Sridhar, J. Linderoth, and J. Luedtke. Locally ideal formulations for piecewise linear functions with indicator variables. *Operations Research Letters*, 41:627–632, 2013.

J. P. Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57:3–57, 2015.

J. P. Vielma, S. Ahmed, and G. Nemhauser. Mixed-integer models for nonseparable piecewise linear optimization: Unifying framework and extensions. *Operations Research*, 58:303–315, 2010.

AA Zoltners and Prabhakant Sinha. Integer programming models for sales resource allocation. *Manag. Sci.*, 26(3):242–260, 1980. URL http://mansci.journal.informs.org/content/26/3/242.short.